米蘭dataset實驗結果

訓練資料與目標：
將dataset依10000 個 square_id區分成100x100的map，並每10分鐘切成一個section，故一天中的資料將會是shape = (100,100,144)的形式，總共有62天的資料

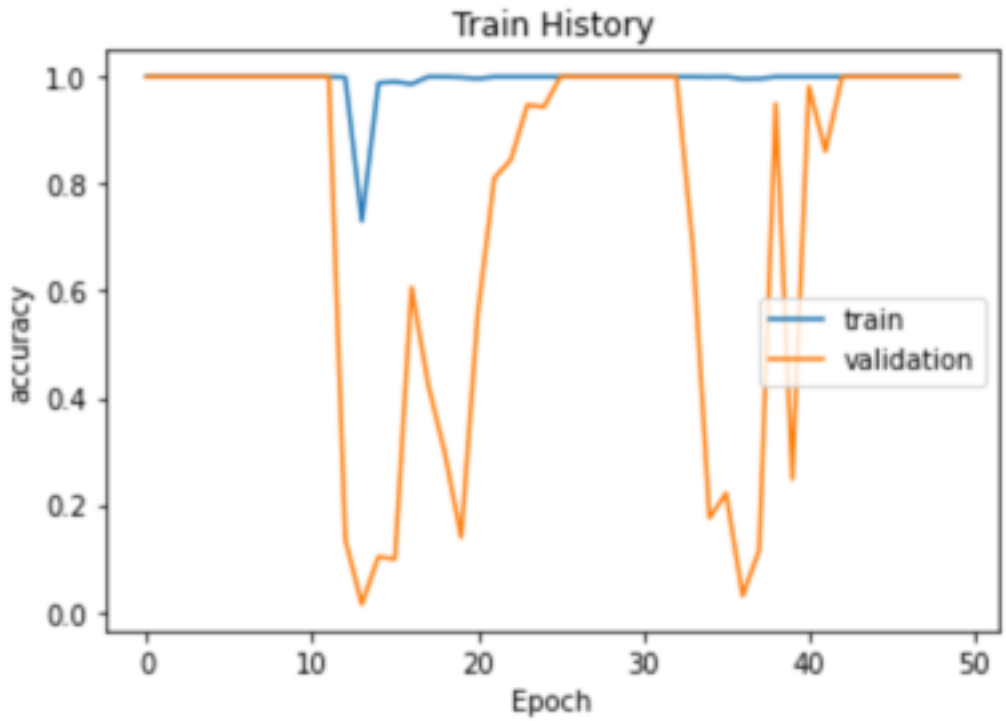
再將144個section每3個section一組，一天分48組，以30分鐘的資料去預測下一個10分鐘的使用熱區位置id
所以將(62,100,100,144)劃分為(2976,100,100,3)，再以4:1:1比例切成train,validation,test，形成訓練data

x_train:(1984,100,100,3)
y_train(1984,)
x_val(496,100,100,3)
y_val(496,)
x_test(496,100,100,3)
y_test(496,)

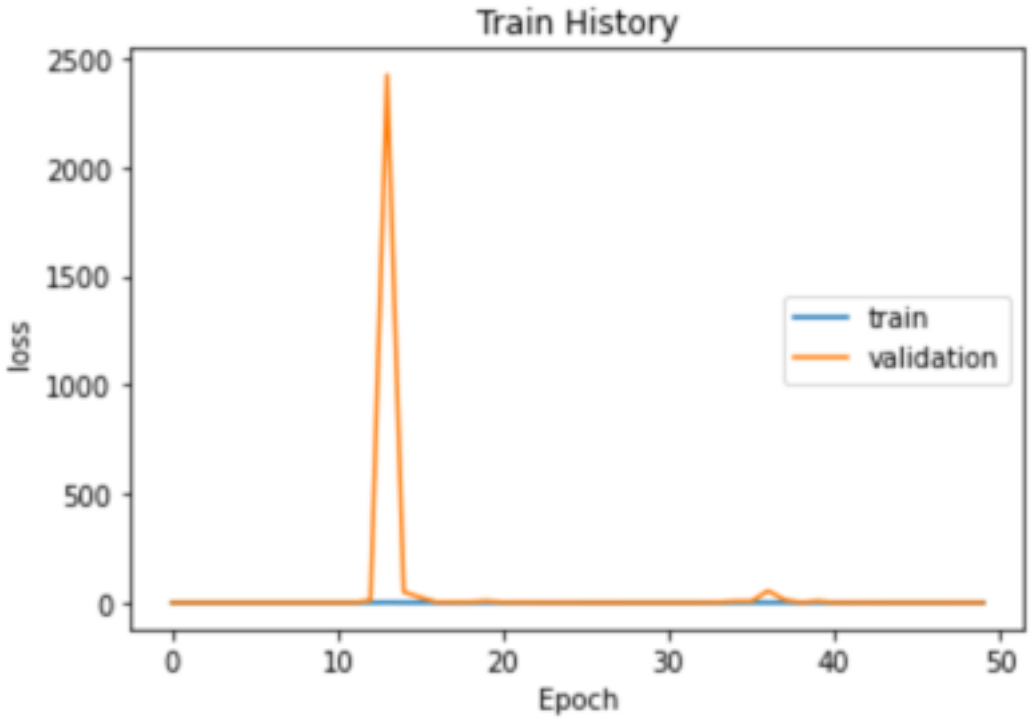
MiniNet實驗結果

```
Epoch 37/50
48/48 [==============================] - 21s 443ms/step - loss: 0.1108 - accuracy: 1.0000 - val_loss: 0.1088 - val_accuracy: 1.0000
Epoch 38/50
48/48 [==============================] - 21s 446ms/step - loss: 0.1087 - accuracy: 1.0000 - val_loss: 0.1079 - val_accuracy: 1.0000
Epoch 39/50
48/48 [==============================] - 21s 444ms/step - loss: 0.1068 - accuracy: 1.0000 - val_loss: 0.1057 - val_accuracy: 1.0000
Epoch 40/50
48/48 [==============================] - 21s 446ms/step - loss: 0.1046 - accuracy: 1.0000 - val_loss: 0.1039 - val_accuracy: 1.0000
Epoch 41/50
48/48 [==============================] - 21s 444ms/step - loss: 0.1026 - accuracy: 1.0000 - val_loss: 0.1019 - val_accuracy: 1.0000
Epoch 42/50
48/48 [==============================] - 21s 447ms/step - loss: 0.1008 - accuracy: 1.0000 - val_loss: 0.0999 - val_accuracy: 1.0000
Epoch 43/50
48/48 [==============================] - 21s 443ms/step - loss: 0.0987 - accuracy: 1.0000 - val_loss: 0.0990 - val_accuracy: 1.0000
Epoch 44/50
48/48 [==============================] - 21s 444ms/step - loss: 0.0970 - accuracy: 1.0000 - val_loss: 0.0978 - val_accuracy: 1.0000
Epoch 45/50
48/48 [==============================] - 21s 444ms/step - loss: 0.0951 - accuracy: 1.0000 - val_loss: 0.0944 - val_accuracy: 1.0000
Epoch 46/50
48/48 [==============================] - 21s 443ms/step - loss: 0.0932 - accuracy: 1.0000 - val_loss: 0.0925 - val_accuracy: 1.0000
Epoch 47/50
48/48 [==============================] - 21s 443ms/step - loss: 0.0915 - accuracy: 1.0000 - val_loss: 0.0910 - val_accuracy: 1.0000
Epoch 48/50
48/48 [==============================] - 21s 444ms/step - loss: 0.0897 - accuracy: 1.0000 - val_loss: 0.0905 - val_accuracy: 1.0000
Epoch 49/50
48/48 [==============================] - 21s 444ms/step - loss: 0.0881 - accuracy: 1.0000 - val_loss: 0.0871 - val_accuracy: 1.0000
Epoch 50/50
48/48 [==============================] - 21s 446ms/step - loss: 0.0863 - accuracy: 1.0000 - val_loss: 0.0855 - val_accuracy: 1.0000
```

```
show_train_history('accuracy','val_accuracy')
```



```
show_train_history('loss','val_loss')
```

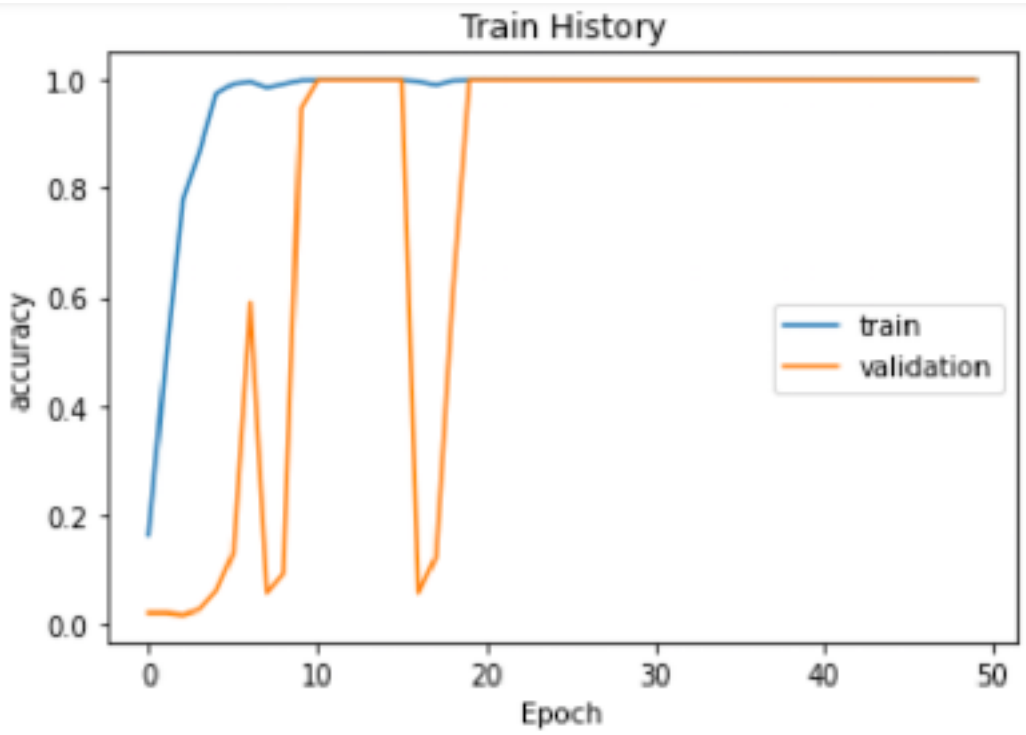



```
[77] score = model.evaluate(x_test,y_test_OneHot, verbose=1)
     print('Test loss:', score[0])
     print('Test accuracy:', score[1])
     predictions = model.predict(x_test, verbose=1)
```

```
16/16 [==============================] - 2s 109ms/step - loss: 0.1562 - accuracy: 0.9980
Test loss: 0.1561591923236847
Test accuracy: 0.9979838728904724
16/16 [==============================] - 2s 93ms/step
```

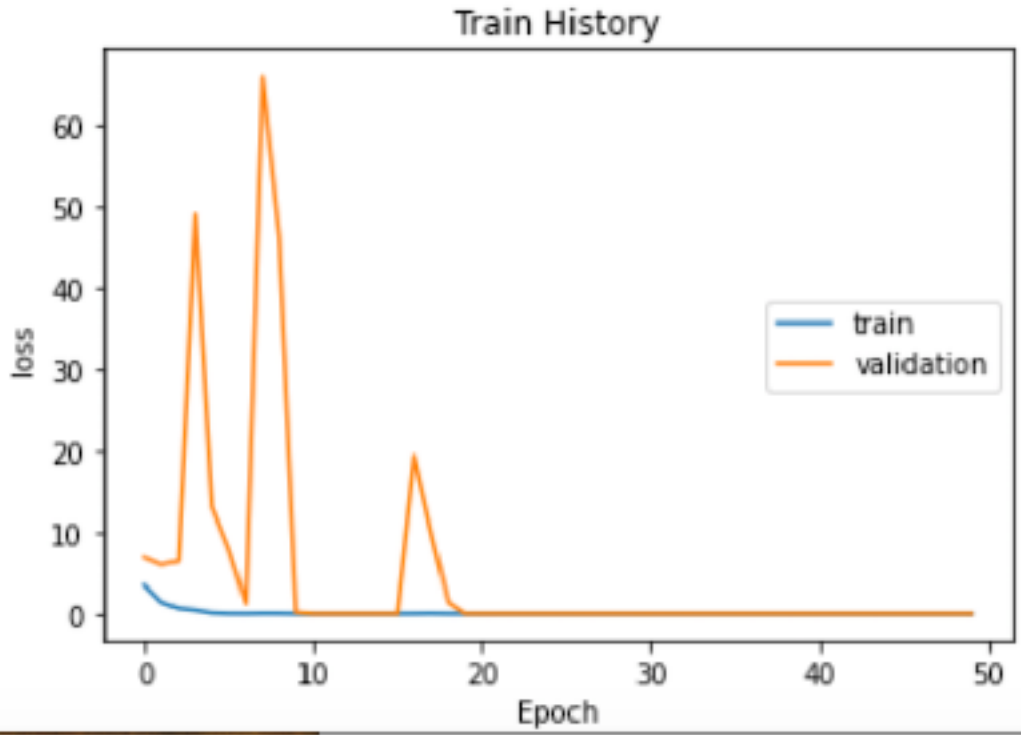
MobileNet實驗結果

```
Epoch 34/50
62/62 [==============================] - 51s 829ms/step - loss: 1.3736e-04 - accuracy: 1.0000 - val_loss: 7.7502e-06 - val_accuracy: 1.0000
Epoch 35/50
62/62 [==============================] - 51s 829ms/step - loss: 8.9340e-05 - accuracy: 1.0000 - val_loss: 7.3548e-06 - val_accuracy: 1.0000
Epoch 36/50
62/62 [==============================] - 51s 827ms/step - loss: 7.8752e-05 - accuracy: 1.0000 - val_loss: 7.0584e-06 - val_accuracy: 1.0000
Epoch 37/50
62/62 [==============================] - 51s 829ms/step - loss: 9.8171e-05 - accuracy: 1.0000 - val_loss: 6.8461e-06 - val_accuracy: 1.0000
Epoch 38/50
62/62 [==============================] - 51s 831ms/step - loss: 1.0365e-04 - accuracy: 1.0000 - val_loss: 6.6004e-06 - val_accuracy: 1.0000
Epoch 39/50
62/62 [==============================] - 50s 813ms/step - loss: 1.0310e-04 - accuracy: 1.0000 - val_loss: 6.1575e-06 - val_accuracy: 1.0000
Epoch 40/50
62/62 [==============================] - 50s 808ms/step - loss: 9.2718e-05 - accuracy: 1.0000 - val_loss: 5.9071e-06 - val_accuracy: 1.0000
Epoch 41/50
62/62 [==============================] - 51s 827ms/step - loss: 9.9461e-05 - accuracy: 1.0000 - val_loss: 5.6890e-06 - val_accuracy: 1.0000
Epoch 42/50
62/62 [==============================] - 51s 827ms/step - loss: 1.2510e-04 - accuracy: 1.0000 - val_loss: 7.1785e-06 - val_accuracy: 1.0000
Epoch 43/50
62/62 [==============================] - 51s 822ms/step - loss: 1.0268e-04 - accuracy: 1.0000 - val_loss: 6.5310e-06 - val_accuracy: 1.0000
Epoch 44/50
62/62 [==============================] - 51s 818ms/step - loss: 9.0622e-05 - accuracy: 1.0000 - val_loss: 6.1779e-06 - val_accuracy: 1.0000
Epoch 45/50
62/62 [==============================] - 51s 827ms/step - loss: 8.9815e-05 - accuracy: 1.0000 - val_loss: 5.6461e-06 - val_accuracy: 1.0000
Epoch 46/50
62/62 [==============================] - 51s 825ms/step - loss: 6.0624e-05 - accuracy: 1.0000 - val_loss: 5.1541e-06 - val_accuracy: 1.0000
Epoch 47/50
62/62 [==============================] - 51s 830ms/step - loss: 8.2691e-05 - accuracy: 1.0000 - val_loss: 4.8203e-06 - val_accuracy: 1.0000
Epoch 48/50
62/62 [==============================] - 51s 827ms/step - loss: 1.0571e-04 - accuracy: 1.0000 - val_loss: 4.8272e-06 - val_accuracy: 1.0000
Epoch 49/50
62/62 [==============================] - 51s 825ms/step - loss: 6.7519e-05 - accuracy: 1.0000 - val_loss: 4.5182e-06 - val_accuracy: 1.0000
Epoch 50/50
62/62 [==============================] - 51s 830ms/step - loss: 8.7598e-05 - accuracy: 1.0000 - val_loss: 4.2922e-06 - val_accuracy: 1.0000
```



In[17]:

[24] show_train_history('loss','val_loss')



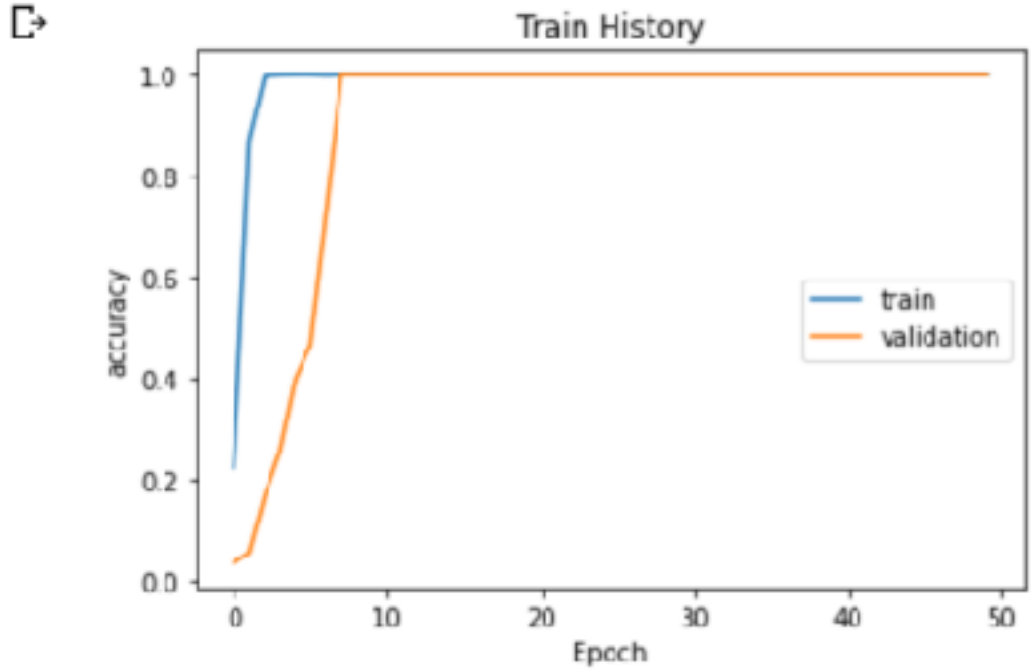

```
score = model.evaluate(x_test,y_test_OneHot, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
predictions = model.predict(x_test, verbose=1)
```

```
16/16 [==============================] - 3s 174ms/step - loss: 0.0801 - accuracy: 0.9980
Test loss: 0.08013397455215454
Test accuracy: 0.9979838728904724
16/16 [==============================] - 3s 174ms/step
```

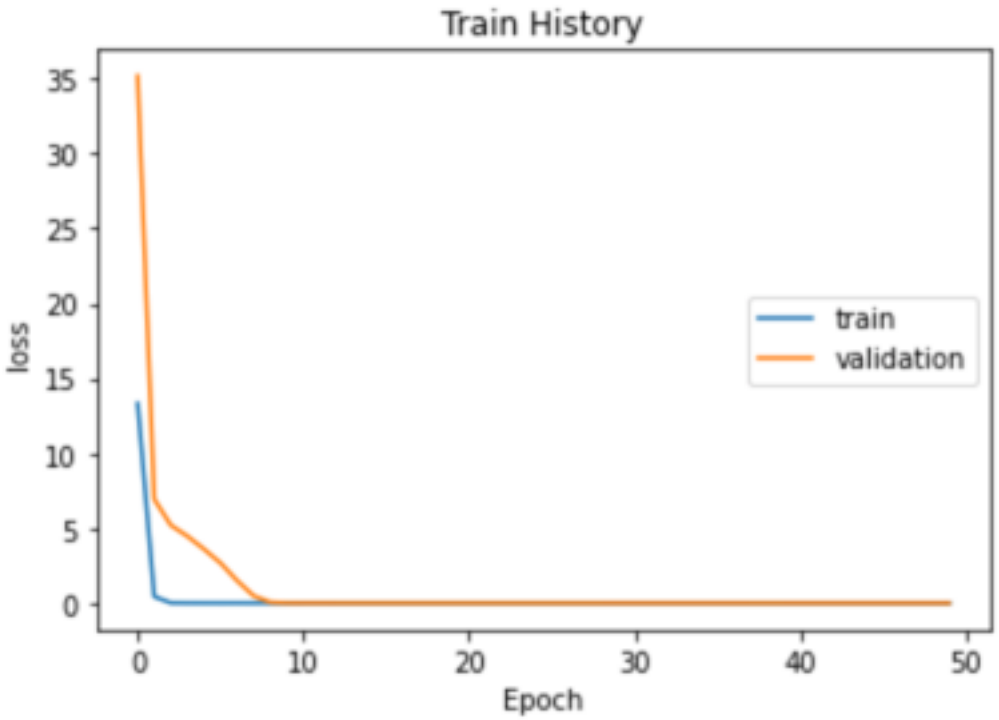
SE_inception_resnet

```
62/62 [==============================] - 68s 1s/step - loss: 2.1369e-04 - accuracy: 1.0000 - val_loss: 7.7713e-05 - val_accuracy: 1.0000
Epoch 36/50
62/62 [==============================] - 68s 1s/step - loss: 2.3565e-04 - accuracy: 1.0000 - val_loss: 7.4405e-05 - val_accuracy: 1.0000
Epoch 37/50
62/62 [==============================] - 68s 1s/step - loss: 2.4812e-04 - accuracy: 1.0000 - val_loss: 7.2850e-05 - val_accuracy: 1.0000
Epoch 38/50
62/62 [==============================] - 68s 1s/step - loss: 2.6958e-04 - accuracy: 1.0000 - val_loss: 7.4737e-05 - val_accuracy: 1.0000
Epoch 39/50
62/62 [==============================] - 67s 1s/step - loss: 5.5395e-04 - accuracy: 1.0000 - val_loss: 1.2192e-04 - val_accuracy: 1.0000
Epoch 40/50
62/62 [==============================] - 68s 1s/step - loss: 2.1580e-04 - accuracy: 1.0000 - val_loss: 8.5209e-05 - val_accuracy: 1.0000
Epoch 41/50
62/62 [==============================] - 68s 1s/step - loss: 2.5124e-04 - accuracy: 1.0000 - val_loss: 7.2664e-05 - val_accuracy: 1.0000
Epoch 42/50
62/62 [==============================] - 68s 1s/step - loss: 2.6221e-04 - accuracy: 1.0000 - val_loss: 6.8105e-05 - val_accuracy: 1.0000
Epoch 43/50
62/62 [==============================] - 68s 1s/step - loss: 2.2319e-04 - accuracy: 1.0000 - val_loss: 8.2458e-05 - val_accuracy: 1.0000
Epoch 44/50
62/62 [==============================] - 68s 1s/step - loss: 2.3143e-04 - accuracy: 1.0000 - val_loss: 6.0590e-05 - val_accuracy: 1.0000
Epoch 45/50
62/62 [==============================] - 68s 1s/step - loss: 1.8808e-04 - accuracy: 1.0000 - val_loss: 5.7923e-05 - val_accuracy: 1.0000
Epoch 46/50
62/62 [==============================] - 68s 1s/step - loss: 2.2765e-04 - accuracy: 1.0000 - val_loss: 5.6675e-05 - val_accuracy: 1.0000
Epoch 47/50
62/62 [==============================] - 68s 1s/step - loss: 1.8587e-04 - accuracy: 1.0000 - val_loss: 5.5797e-05 - val_accuracy: 1.0000
Epoch 48/50
62/62 [==============================] - 68s 1s/step - loss: 2.6068e-04 - accuracy: 1.0000 - val_loss: 5.6887e-05 - val_accuracy: 1.0000
Epoch 49/50
62/62 [==============================] - 68s 1s/step - loss: 1.4527e-04 - accuracy: 1.0000 - val_loss: 5.0968e-05 - val_accuracy: 1.0000
Epoch 50/50
62/62 [==============================] - 68s 1s/step - loss: 2.3161e-04 - accuracy: 1.0000 - val_loss: 5.2033e-05 - val_accuracy: 1.0000
```

```
show_train_history('accuracy','val_accuracy')
```



```
show_train_history('loss','val_loss')
```



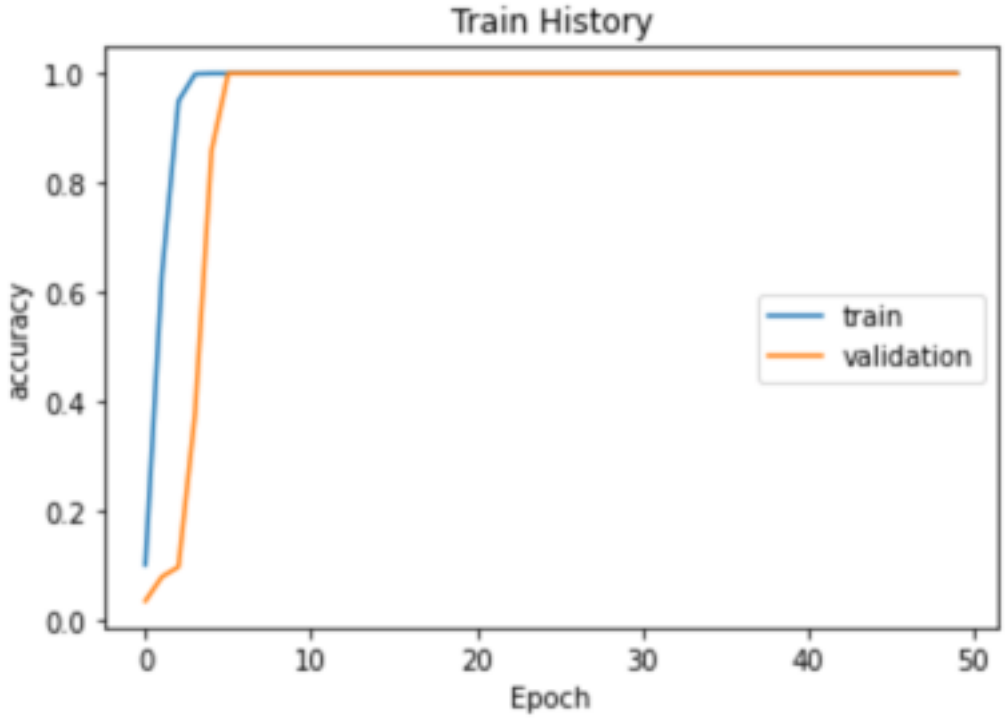

```
[29] score = model.evaluate(x_test,y_test_OneHot, verbose=1)
     print('Test loss:', score[0])
     print('Test accuracy:', score[1])
     predictions = model.predict(x_test, verbose=1)
```

```
16/16 [==============================] – 4s 275ms/step – loss: 0.0527 – accuracy: 0.9980
Test loss: 0.05267724022269249
Test accuracy: 0.9979838728904724
16/16 [==============================] – 5s 272ms/step
```

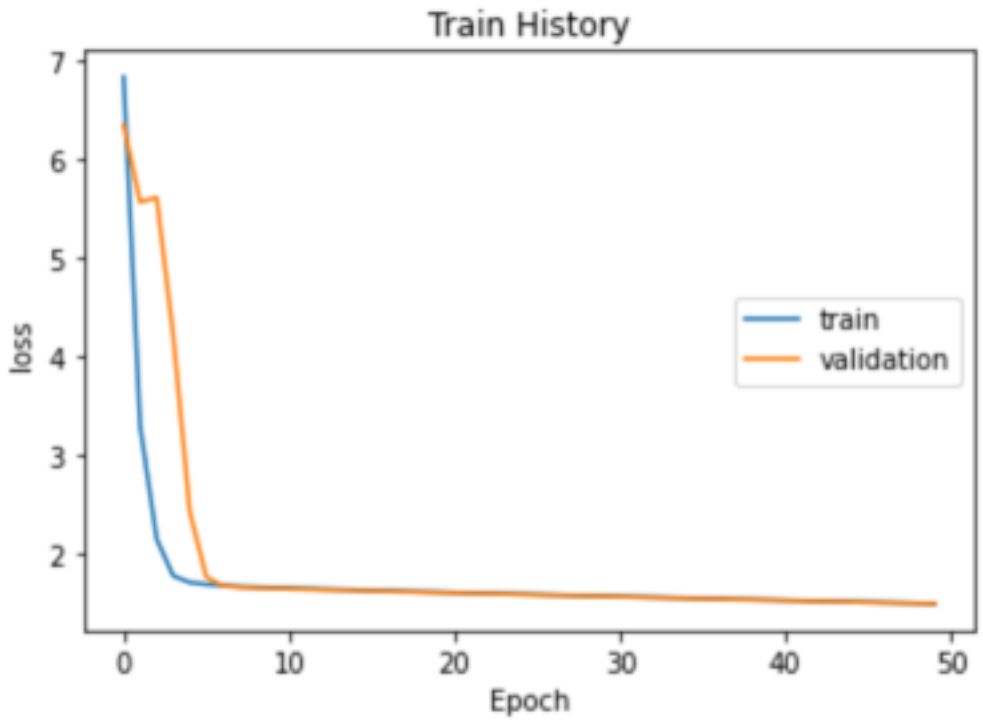
Densenet:

```
Epoch 34/50
62/62 [==============================] - 67s 1s/step - loss: 1.5449 - accuracy: 1.0000 - val_loss: 1.5409 - val_accuracy: 1.0000
Epoch 35/50
62/62 [==============================] - 67s 1s/step - loss: 1.5404 - accuracy: 1.0000 - val_loss: 1.5370 - val_accuracy: 1.0000
Epoch 36/50
62/62 [==============================] - 66s 1s/step - loss: 1.5364 - accuracy: 1.0000 - val_loss: 1.5332 - val_accuracy: 1.0000
Epoch 37/50
62/62 [==============================] - 67s 1s/step - loss: 1.5326 - accuracy: 1.0000 - val_loss: 1.5294 - val_accuracy: 1.0000
Epoch 38/50
62/62 [==============================] - 66s 1s/step - loss: 1.5287 - accuracy: 1.0000 - val_loss: 1.5256 - val_accuracy: 1.0000
Epoch 39/50
62/62 [==============================] - 66s 1s/step - loss: 1.5250 - accuracy: 1.0000 - val_loss: 1.5219 - val_accuracy: 1.0000
Epoch 40/50
62/62 [==============================] - 67s 1s/step - loss: 1.5211 - accuracy: 1.0000 - val_loss: 1.5181 - val_accuracy: 1.0000
Epoch 41/50
62/62 [==============================] - 67s 1s/step - loss: 1.5174 - accuracy: 1.0000 - val_loss: 1.5143 - val_accuracy: 1.0000
Epoch 42/50
62/62 [==============================] - 66s 1s/step - loss: 1.5137 - accuracy: 1.0000 - val_loss: 1.5105 - val_accuracy: 1.0000
Epoch 43/50
62/62 [==============================] - 66s 1s/step - loss: 1.5097 - accuracy: 1.0000 - val_loss: 1.5068 - val_accuracy: 1.0000
Epoch 44/50
62/62 [==============================] - 67s 1s/step - loss: 1.5061 - accuracy: 1.0000 - val_loss: 1.5031 - val_accuracy: 1.0000
Epoch 45/50
62/62 [==============================] - 66s 1s/step - loss: 1.5024 - accuracy: 1.0000 - val_loss: 1.4993 - val_accuracy: 1.0000
Epoch 46/50
62/62 [==============================] - 67s 1s/step - loss: 1.4985 - accuracy: 1.0000 - val_loss: 1.4956 - val_accuracy: 1.0000
Epoch 47/50
62/62 [==============================] - 67s 1s/step - loss: 1.4947 - accuracy: 1.0000 - val_loss: 1.4919 - val_accuracy: 1.0000
Epoch 48/50
62/62 [==============================] - 66s 1s/step - loss: 1.4911 - accuracy: 1.0000 - val_loss: 1.4882 - val_accuracy: 1.0000
Epoch 49/50
62/62 [==============================] - 67s 1s/step - loss: 1.4874 - accuracy: 1.0000 - val_loss: 1.4845 - val_accuracy: 1.0000
Epoch 50/50
62/62 [==============================] - 66s 1s/step - loss: 1.4837 - accuracy: 1.0000 - val_loss: 1.4808 - val_accuracy: 1.0000
```

```
[22] show_train_history('accuracy','val_accuracy')
```



```
show_train_history('loss','val_loss')
```




```
score = model.evaluate(x_test,y_test_OneHot, verbose=1)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
predictions = model.predict(x_test, verbose=1)
```

```
16/16 [==============================] – 4s 266ms/step – loss: 1.5291 – accuracy: 0.9980
Test loss: 1.5290913581848145
Test accuracy: 0.9979838728904724
16/16 [==============================] – 6s 250ms/step
```

實驗觀察結果：

四種model在訓練時雖然有的偶爾會出現loss及accuracy大波動的情形，但大多數情況都能夠穩定收在近乎100%的準確率，雖然loss也確實有在持續降低，但也不確定這麼高到底是不是真的可以找到一套預測參數，還是有什麼地方可以偷吃步
目前都盡量調整跟學姊的輸入輸出格式相仿，模型架構也都沒有更動，僅調整class、input_size這些變數